Supplementary Material of "Temporal-Spectral Shifted Diffusion Models for Time Series Forecasting"

Anonymous Authors[1]

A. Theoretical Derivations

A.1. The Distributions in The Forward Process

First, we derive the explicit expressions for $q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ and $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$, based on our shifted diffusion

Lemma A.1. *For the forward process $q(\boldsymbol{z}_1^{tar}, \boldsymbol{z}_2^{tar}, ..., \boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) = \prod_{t=1}^{T} q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$, if the transition kernel $q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ is defined as **??**, then the conditional distribution $q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ has the desired distribution as **??**, i.e., $\mathcal{N}(\boldsymbol{z}_t^{tar}; \boldsymbol{z}_{t-1}^{tar} + k_t s_\phi(\boldsymbol{z}^c, v_s), k_t \boldsymbol{I})$.*

Proof. We prove the lemma by induction. Suppose at time t, we have $q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ and $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ admit the desired distributions as in **????**, respectively, then we need to prove that $q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) = \mathcal{N}(\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar} + k_t s_\phi(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, t), k_t \boldsymbol{I})$. We can re-write he conditional distributions of $\boldsymbol{z}_t^{tar}$ given $(\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ and $\boldsymbol{z}_{t-1}^{tar}$ given $(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$ with the following equations:

$$\begin{aligned} \boldsymbol{z}_t^{tar} &= \boldsymbol{z}_{t-1}^{tar} + \alpha_t s_\phi(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) + \sqrt{\alpha_t}\epsilon_{t-1}, \\ &= \boldsymbol{z}_0^{tar} + \Sigma_{i=1}^{t} \alpha_t s_\phi(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) + \Sigma_{i=1}^{t} \sqrt{\alpha_t}\epsilon_i \end{aligned} \tag{1}$$

where ϵ_i are independent standard gaussian random variables. We can simplify Equation (1) as:

$$\boldsymbol{z}_t^{tar} = \boldsymbol{z}_0^{tar} + k_t s_\phi(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) + \sqrt{k_t}\epsilon_t \tag{2}$$

Then the marginal distribution of **??** is obtained from Equation (2). □

Proposition A.2. *Suppose the distribution of forward process is defined by **????**, then at each time t, the posterior distribution $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)$ is described by **??***

Proof. By the Bayes rule, $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c) = \frac{q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c) q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)}{q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)}$. By **????**, the numerator and denominator are both gaussian , then the posterior distribution is also gaussian and we can proceed to calculate its mean and variance:

$$\begin{aligned} q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c) = &\frac{\mathcal{N}(\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar} + k_{t-1}\hat{s}, k_{t-1}\boldsymbol{I})}{\mathcal{N}(\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar} + k_t\hat{s}, k_t\boldsymbol{I})} \quad , \\ &* \mathcal{N}(\boldsymbol{z}_t^{tar}, \boldsymbol{z}_{t-1}^{tar} + \alpha_t\hat{s}, \alpha_t\boldsymbol{I}) \end{aligned} \tag{3}$$

where $\hat{s}$ is an abbreviation form of $k_t s_\phi(\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s)$. Dropping the constants which are unrelated to $\boldsymbol{z}_0^{tar}, \boldsymbol{z}_t^{tar}, \boldsymbol{z}_{t-1}^{tar}$ and $\boldsymbol{z}^c$, we have:

$$\begin{aligned} q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c) &\propto exp\left\{ -\frac{(\boldsymbol{z}_{t-1}^{tar} - \boldsymbol{z}_0^{tar} - k_{t-1}\hat{s})^2}{2k_{t-1}} + \frac{(\boldsymbol{z}_t^{tar} - \boldsymbol{z}_0^{tar} - k_t\hat{s})^2}{2k_t} \right. \\ &\left. - \frac{(\boldsymbol{z}_t^{tar} - \boldsymbol{z}_{t-1}^{tar} - \alpha_t\hat{s})^2}{2\alpha_t} \right\} \\ &= exp\left\{ C(\boldsymbol{z}_0^{tar}, \boldsymbol{z}_t^{tar}, \boldsymbol{z}^c) - \frac{1}{2}(\frac{1}{k_{t-1}} + \frac{1}{\alpha_t}) * (\boldsymbol{z}_{t-1}^{tar})^2 + \boldsymbol{z}_{t-1}^{tar} * \right. \\ &\left. [\frac{(\boldsymbol{z}_0^{tar} + k_{t-1}\hat{s})}{k_{t-1}} + \frac{(\boldsymbol{z}_t^{tar} - \alpha_t\hat{s})}{\alpha_t}] \right\} \\ &= exp\left\{ C(\boldsymbol{z}_0^{tar}, \boldsymbol{z}_t^{tar}, \boldsymbol{z}^c) - \frac{1}{2}(\frac{1}{k_{t-1}} + \frac{1}{\alpha_t}) * (\boldsymbol{z}_{t-1}^{tar})^2 + \boldsymbol{z}_{t-1}^{tar} * \right. \\ &\left. (\frac{1}{k_{t-1}}\boldsymbol{z}_0^{tar} + \frac{1}{\alpha_t}\boldsymbol{z}_t^{tar}]) \right\}, \end{aligned} \tag{4}$$

where $C(\boldsymbol{z}_0^{tar}, \boldsymbol{z}_t^{tar}, \boldsymbol{z}^c)$ is a constant term with respect to $\boldsymbol{z}_{t-1}^{tar}$. Note that $(\frac{1}{k_{t-1}} + \frac{1}{\alpha_t}) = \frac{k_t}{k_{t-1}}$, and with some algebraic derivation, we can show that the gaussian distribution $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)$ has:

$$\begin{aligned} variance: &\frac{k_{t-1}}{k_t}\alpha_t\boldsymbol{I} \\ mean: &\frac{\alpha_t}{k_t}\boldsymbol{z}_0^{tar} + \frac{k_{t-1}}{k_t}\boldsymbol{z}_t^{tar} \end{aligned} \tag{5}$$

□

This quadratic form induces the Gaussian distribution of **??**

A.2. Upper Bound of The Likelihood

Here we show with our parameterization, that the objective function $\mathcal{L}_{\theta,\phi}$ **??** is an upper bound of the negative log-likelihood of the data distribution.

Lemma A.3. *Based on the non-Markovian forward process* $q(\boldsymbol{z}_1^{tar}, \boldsymbol{z}_2^{tar}, ..., \boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c) = \prod_{t=1}^T q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)$ *and the conditional reverse process* $p_\theta(\boldsymbol{z}_0^{tar}, \boldsymbol{z}_1^{tar}, \boldsymbol{z}_2^{tar}, ..., \boldsymbol{z}_T^{tar}|\boldsymbol{z}^c) = p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c)\prod_{t=1}^T p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^c)$*, the objective function* **??** *is an upper bound of the negative log likelihood.*

Proof.

$$
\begin{aligned}
-\log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}^c) &\le -\log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}^c) + \mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{-\log\frac{p_\theta(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= \mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{-\log\frac{p_\theta(\boldsymbol{z}_{0:T}^{tar}|\boldsymbol{z}^c)}{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= -\mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{\log\frac{p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c)\prod_{t=1}^T p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)}{\prod_{t=1}^T q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= -\mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{\log p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c) + \sum_{t>1}\log\frac{p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)} + \log\frac{p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_1^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= -\mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{\log p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c) + \log\frac{p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_1^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right. \\
&\quad \left. + \sum_{t>1}\log\frac{p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)} * \frac{q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= -\mathbb{E}_{q(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\left\{\log\frac{p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c)}{q(\boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)} + \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar},\boldsymbol{z}^c) + \log\sum_{t>1}\frac{p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)}{q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\right\} \\
&= D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)\|p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c)) - \mathbb{E}_{q(\boldsymbol{z}_1^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar},\boldsymbol{z}^c) \\
&\quad + \sum_{t>1}\mathbb{E}_{q(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)\|p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c))
\end{aligned}
\tag{6}
$$

□

Assume that the total diffusion step T is big enough and only a negligible amount of noise is added to the data at the first diffusion step, then the term $D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)\|p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c)) - \mathbb{E}_{q(\boldsymbol{z}_1^{tar}|\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)}\log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar},\boldsymbol{z}^c)$ is approximately zero. Now with Lemma A.3, we have the following proposition:

Proposition A.4. *The objective function defined in* **??** *is an upper bound of the negative log-likelihood.*

A.3. Achieving Better Likelihood with TF-shifter

Next, we show that the TF-SHIFTER is theoretically capable of achieving better likelihood compared to original DDPMs. As the exact likelihood is intractable, we aim to compare the optimal variational bounds for negative log-likelihoods. The objective function of TF-SHIFTER at time step t is $E_{q_\phi}D_{KL}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c))$, and its optimal solution is

$$
\begin{aligned}
&\min_{\phi,\theta}\mathbb{E}_{q_\phi}D_{KL}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)) \\
&= min_\phi[min_\theta\mathbb{E}_{q_\phi}D_{KL}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c))] \\
&\le min_\theta\mathbb{E}_{q_{\phi=0}}D_{KL}(q_{\phi=0}(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}_0^{tar},\boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar},\boldsymbol{z}^c)),
\end{aligned}
\tag{7}
$$

Table 1. Summary of dataset statistics, including dimension, total observations, sampling frequency, and prediction length used in the experiments.

dataset	dim	#observations	freq.	H (steps)
Weather	21	52,696	10 mins	1 week (672)
ETTm1	7	69,680	15 mins	2 days (192)
Wind	7	48,673	15 mins	2 days (192)
Traffic	862	17,544	1 hour	1 week (168)
Electricity	321	26,304	1 hour	1 week (168)
Exchange	8	7,588	1 day	2 weeks (14)

where $\phi = 0$ denotes setting the adapter network identical to 0, and thus $min_\theta \mathbb{E}_{q_{\phi=0}} D_{KL}(q_{\phi=0}(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^c))$ is the optimal loss of origianl DDPMs objective at time t. Similar inequality can be obtained for t=1:

$$\begin{aligned} &\min_{\phi,\theta} \mathbb{E}_{q_\phi} - \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar}, \boldsymbol{z}^c) \\ &\leq \min_{\theta} \mathbb{E}_{q_{\phi=0}} - \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar}, \boldsymbol{z}^c). \end{aligned} \tag{8}$$

As a result, we have the following inequality by summing up the objectives at all time steps:

$$\begin{aligned} &- \mathbb{E}_{q(\boldsymbol{z}_0^{tar})} \log p_\theta(\boldsymbol{z}_0^{tar}) \\ &\leq min_{\phi,\theta} \sum_{t>1} \mathbb{E}_{q_\phi} D_{KL}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^c)) + \mathbb{E}_{q_\phi} - \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar}, \boldsymbol{z}^c) + C \\ &\leq min_{\theta} \sum_{t>1} \mathbb{E}_{q_{\phi=0}} D_{KL}(q_{\phi=0}(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)||p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^c)) + \mathbb{E}_{q_{\phi=0}} - \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar}, \boldsymbol{z}^c) + C \end{aligned} \tag{9}$$

, where $C = \mathbb{E} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c)\|p_\theta(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^c))$ is a constant. Hence, the TF-SHIFTER has a tighter bound for the Negative Log-Likelihood (NLL), and is thus theoretically capable of achieving more accurate distribution, compared with the original DDPMs.

B. Implementation Details

B.1. Datasets Details

Due to varying sampling interval lengths across different datasets, employing the fixed set of prediction horizons 96, 192, 336, 720 for all datasets, as in (**??**), may not be appropriate. For instance, the *Exchange* dataset consists of daily exchange rates. A prediction horizon of 720 corresponds to forecasting two years into the future, which may be excessive. Instead, we set the prediction horizon H to 14, corresponding to 2 weeks into the future, which is deemed more reasonable. Similarly, we set H = 168 for *ETTm1* (equivalent to 1 week into the future), H =672 for *Weather* (corresponding to 1 week), and so forth, as outlined in Table 1. It is noteworthy that some papers also choose the prediction length based on the dataset's sampling frequency. For example, **?** (**?**) and **?** (**?**) also use 168 (instead of 192) for *ETTm1* and *Traffic*.

In the diffusion process, we use the following 128-dimensions time-step embedding following previous works (**???**):

$$t_{embedding}(t) = \left(\sin(10^{0\cdot 4/63}t), \ldots, \sin(10^{63\cdot 4/63}t), \cos(10^{0\cdot 4/63}t), \ldots, \cos(10^{63\cdot 4/63}t)\right). \tag{10}$$

B.2. Network Architecture

STFT, ISTFT, and Convolutional Kernel and Stride Sizes. The STFT (Short-Time Fourier Transform) and ISTFT (Inverse Short-Time Fourier Transform) are implemented using [1] and [2], respectively. The window length of the short-time Fourier transform is set to 8. Under this setting, the range of the frequency axis is [1, 2, 3, 4, 5]. Due to the default setting of the hop length as 2, the length of the time-frequency spectrogram becomes half of the given time series. The encoder

[1] https://pytorch.org/docs/stable/generated/torch.stft.html
[2] https://pytorch.org/docs/stable/generated/torch.istft.html

consists of multiple downsampling layers, compressing the input into the latent space. The downsampling rate is determined by the convolutional kernel and stride size, typically using 2D Convolution layer(kernel size: 4, stride: 2, padding: 1) for a 2x downsampling at each downsampling layer. However, selecting these sizes is challenging due to variations in the time and frequency axis lengths across different datasets. Therefore, we let the downsampling layers only downsample along the time axis, rather than the frequency axis. Subsequently, the downsampling layers are transformed into 2D convolutional layers (kernel size: (3, 4), stride: (1, 2), padding: (1, 1)).

Our experiments found that smaller STFT windows, such as 4 or 8, achieve better performance. This is related to the amount of compression applied to the input data. Larger STFT windows result in a wider frequency axis and a shorter time axis, reducing the amount of compression as downsampling is applied only along the time axis. The global coherence of generated samples is often poorer at larger compression amounts (i.e., lower downsampling rates). We only reconstruct the prediction part and use all frequency bands to avoid reconstruction errors.

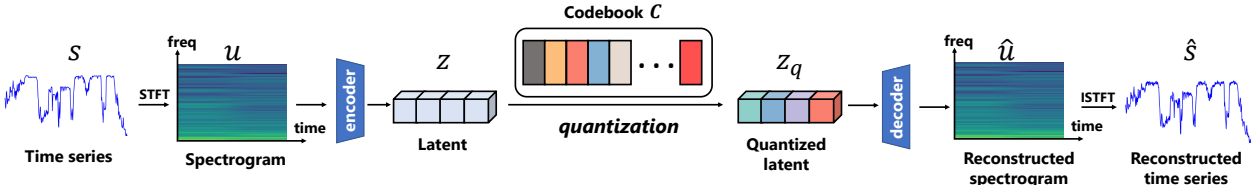

Figure 1. The framework of training VQ-VAE. We train VQ-VAE by minimizing the distance between reconstructed time series $\hat{X}$ and original time series X, as well as the distance between the reconstructed spectrogram $\hat{u}$ and the original spectrogram u.

VQ-VAE. We train the VQ-VAE using the framework depicted in 1. In our approach, we employ the same encoder and decoder architecture as outlined in the VQ-VAE paper, with their implementations referenced from (**?**). The encoder consists of 4 downsampling convolutional blocks (Conv2d – BatchNorm2d – LeakyReLU) followed by 4 residual blocks (LeakyReLU – Conv2d – BatchNorm2d – LeakyReLU – Conv2d). The downsampling convolutional layers and residual convolutional layers are implemented using two-dimensional convolutional layers (kernel size=(3,4), stride=(1,2), padding=(1,1)) and (kernel size=(3,3), stride=(1,1), padding=(1,1)), respectively. Prior to encoding, additional padding modules are introduced to prevent label leakage and shape difference between output and input.

The decoder similarly comprises 4 residual blocks followed by 4 upsampling convolutional blocks, mirroring the network details of the encoder. The codebook size K is set to 512, and the code dimension size aligns with the hidden dimension size of the encoder and decoder. For codebook learning loss, we adopt an alternative approach involving commitment loss and exponential moving average, as mentioned in (**?**). We set β to 1, and the exponential moving average decay rate to 0.8.

Transformers. Our approach employs the Transformer architecture from CSDI, with the distinction of expanding the channel dimension to 128. The network comprises temporal and feature layers, ensuring the comprehensiveness of the model in handling the time-frequency domain latent while maintaining a relatively simple structure. Regarding the transformer layer, we utilized a 1-layer Transformer encoder implemented in PyTorch (**?**), comprising multi-head attention layers, fully connected layers, and layer normalization. We adopted the "linear attention transformer" package [3], to enhance computational efficiency. The inclusion of numerous features and long sequences prompted this decision. The package implements an efficient attention mechanism (**?**), and we exclusively utilized the global attention feature within the package.

Our shift network is even smaller, consisting of a transformer with only 4 heads and 32 dimensions. Its role is more about transmitting observed information to the shift while ensuring dimensional alignment between observation and targets.

Side Information. We utilize the combination of temporal embedding and feature embedding as side information v_s. We use 128-dimensions temporal embedding following previous studies (**?**):

$$s_{embedding}(s_l) = \left(\sin(s_l/\tau^{0/64}), \ldots, \sin(s_l/\tau^{63/64}), \cos(s_l/\tau^{0/64}), \ldots, \cos(s_l/\tau^{63/64})\right) \tag{11}$$

where $\tau = 10000$. Following (**?**), s_l represents the timestamp corresponding to the l-th point in the time series. This setup is designed to capture the irregular sampling in the dataset and convey it to the model. Additionally, we utilize learnable embedding to handle feature dimensions. Specifically, feature embedding is represented as 16-dimensional learnable vectors that capture relationships between dimensions. According to (**?**), we combine time embedding and feature embedding, collectively referred to as side information v_s.

[3] https://github.com/lucidrains/linear-attention-transformer

The shape of v_s is not fixed and varies with datasets. Taking the Exchange dataset as an example, the shape of forecasting target Y is [Batchsize (64), 7(number of variables), 168 (time-dimension), 12 (time-dimension)] and the corresponding shape of v_s is [Batchsize (64), total channel(144(time:128 + feature:16)), 320 (frequency-dimension*latent channel), 12 (time-dimension)], and the shape of shift is [Batchsize (64), 64(latent channels), 5 (frequency-dimension), 12 (time-dimension)].

B.3. Baselines

Codes for the baselines are downloaded from the following. (i) TimeGrad: https://github.com/ForestsKing/TimeGrad; (ii) CSDI: https://github.com/ermongroup/CSDI; (iii) SSSD: https://github.com/AI4HealthUOL/SSSD; (iv) D^3VAE: https://github.com/ramber1836/d3vae; (v) FiLM: https://github.com/DAMO-DI-ML/NeurIPS2022-FiLM; (vi) Depts: https://github.com/weifantt/DEPTS; (vii) NBeats: https://github.com/ServiceNow/N-BEATS; (viii) SCINet: https://github.com/cure-lab/SCINet; (ix) Fedformer: https://github.com/DAMO-DI-ML/ICML2022-FEDformer; (x) Autoformer: https://github.com/thuml/Autoformer; (xi) Pyraformer: https://github.com/ant-research/Pyraformer; (xii) Informer: https://github.com/zhouhaoyi/Informer2020; (xiii) Transformer: https://github.com/thuml/Autoformer/blob/main/models/Transformer.py; (xiv) DLinear: https://github.com/ioannislivieris/DLinear; (xv) LSTM-a: https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html. (xvi) FreTS: https://github.com/aikunyi/FreTS.

B.4. Metrics

We will introduce the metrics in our experiments. We summarize them as below:

CRPS. CRPS (**?**) is a univariate strictly proper scoring rule which ´ measures the compatibility of a cumulative distribution function F with an observation x as:

$$CRPS(F, x) = \int_R (F(y) - \mathbb{1}_{(x \le y)})^2 dy \tag{12}$$

where $\mathbb{1}_{(x \le y)}$ is the indicator function, which is 1 if $x \le y$ and 0 otherwise. The CRPS attains the minimum value when the predictive distribution F same as the data distribution.

QICE. To enhance the assessment of uncertainty estimation capabilities in probabilistic multivariate time series forecasting tasks, we introduce Quantile Interval Coverage Error (QICE) (**?**). With a sufficient number of $y_{0:M}$ samples, the first step involves dividing them into M quantile intervals (QIs), each with approximately equal sizes. Subsequently, quantile values corresponding to each QI boundary are determined:

$$QICE = \frac{1}{M}\sum_{m=1}^{M} |r_m - \frac{1}{M}|, \text{where } r_m = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{y_n \ge \hat{y_{nn}}^{low_m}} \cdot \mathbb{1}_{y_n \le \hat{y_{nn}}^{high_m}} \tag{13}$$

where $\hat{y_{nn}}^{low_m}$ and $\hat{y_{nn}}^{low_m}$ represent the low and high percentiles, respectively, of our choice for the predicted $y_{0:M}$ outputs given the input. In cases where the learned distribution accurately represents the true distribution, this measurement should closely align with the difference between the selected low and high percentiles.

MAE and MSE. MAE and MSE are calculated in the formula below, $\hat{Y}$ represents the predicted time series, and Y represents the ground truth time series. MAE calculates the average absolute difference between predictions and true values, while MSE calculates the average squared difference between predictions and true values. A smaller MAE or MSE implies better predictions.

$$\begin{aligned} MAE &= mean(|\hat{Y} - Y|) \\ MSE &= \sqrt{mean(|\hat{Y} - Y|)} \end{aligned} \tag{14}$$

C. More Experiment Results

Table 2 shows the complete results of **??**.

Table 2. The complete results of QICE/CRPS on six real world datasets.

Dataset	Exchange		Wind		Electricity		Weather		Traffic		ETTm1	
Method	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS
TF-SHIFTER (ours)	**4.332**	**0.339**	**6.980**	**0.920**	**5.291**	0.397	**3.814**	**0.324**	3.924	0.299	**3.742**	**0.374**
TimeGrad	4.479	0.590	7.567	0.923	5.396	0.491	7.302	0.411	3.784	0.369	5.374	0.605
CSDI	4.337	0.397	7.716	0.941	5.401	0.480	5.156	0.354	3.429	0.360	5.029	0.482
SSSD	5.121	0.427	7.601	0.981	5.711	0.429	9.409	0.350	3.792	0.401	4.864	0.556
D_3VAE	10.356	0.401	11.118	0.979	13.593	**0.389**	12.985	0.381	12.928	0.483	12.162	0.431
Fedformer	6.623	0.631	11.093	1.235	7.468	0.561	6.262	0.347	3.821	0.505	4.315	0.503
FreTS	7.839	0.440	10.004	0.943	8.139	0.634	6.471	0.354	8.512	0.602	4.437	0.522
FiLM	9.014	0.349	7.569	6.998	9.969	0.671	5.849	0.336	4.270	0.478	3.971	0.474
Autoformer	10.256	0.769	9.896	1.026	6.513	0.602	8.452	0.354	4.368	0.463	5.312	0.566
Pyraformer	11.432	0.532	9.532	0.994	7.432	0.732	10.369	0.385	4.119	0.460	6.311	0.505
Informer	10.173	0.631	12.031	1.065	12.351	0.749	9.411	0.364	4.204	0.591	8.936	0.631
Transformer	8.162	0.629	12.221	1.026	13.627	0.801	10.317	0.370	5.011	0.630	12.191	0.742
SCINet	10.831	0.624	11.125	0.997	5.642	0.499	6.262	0.344	4.011	0.505	6.139	0.531
DLinear	5.017	0.538	7.613	0.957	5.998	0.527	9.311	0.444	3.919	0.318	4.915	0.498
NLinear	5.931	0.481	9.019	0.974	5.510	0.419	6.131	0.328	5.161	0.373	4.62	0.495
Depts	5.058	0.520	9.113	1.001	8.357	0.803	11.132	0.394	12.519	0.712	4.917	0.527
NBeats	4.516	0.399	11.131	0.981	6.931	0.697	12.993	0.420	**3.420**	**0.291**	5.031	0.519

To emphasize our distribution estimation capabilities, we present the predicted median and visualize the 50% and 90% distribution intervals in Figure 2. We compare TF-SHIFTER with CSDI and TimeGrad. Clearly, our method has achieved the best results, while the other two methods performed poorly on the forecasting task with a large horizon.

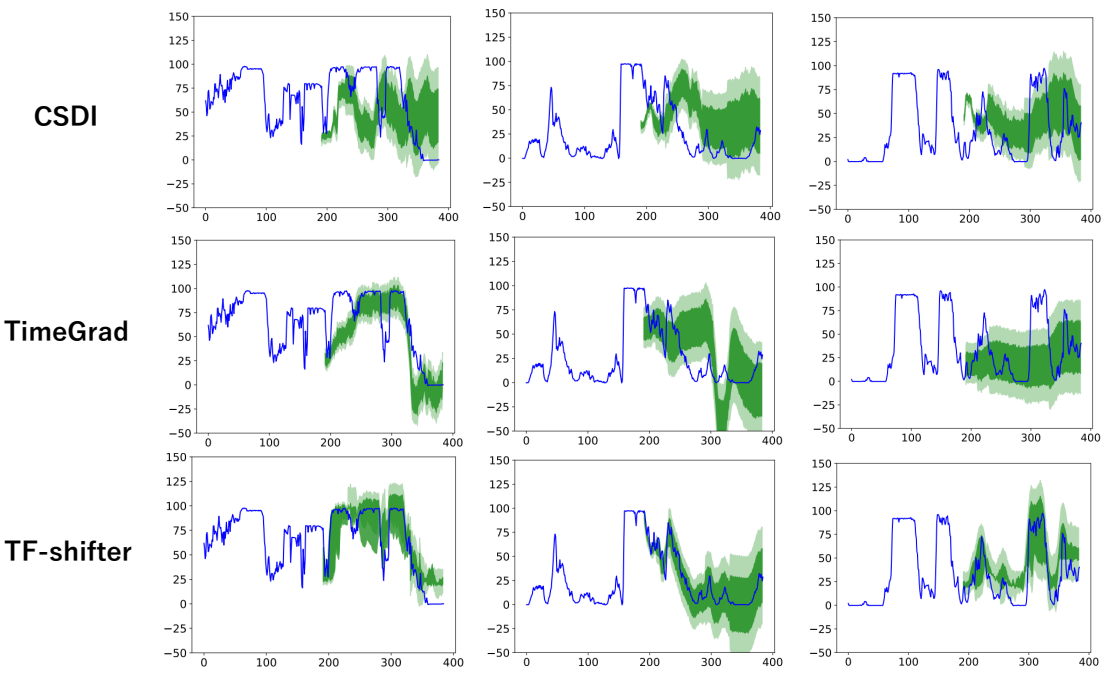

Figure 2. Comparison of prediction intervals for the Wind dataset. We display the predicted median and visualize the 50% and 90% distribution intervals, the blue line representing the test set ground truth.

3 represents the comparison between our predictions on the strongly periodic dataset electricity and those of CSDI. It can be observed that our predictions are significantly better compared to CSDI. Moreover, our method demonstrates a clear ability to accurately forecast the periodicity of time series.

D. More Model Analysis

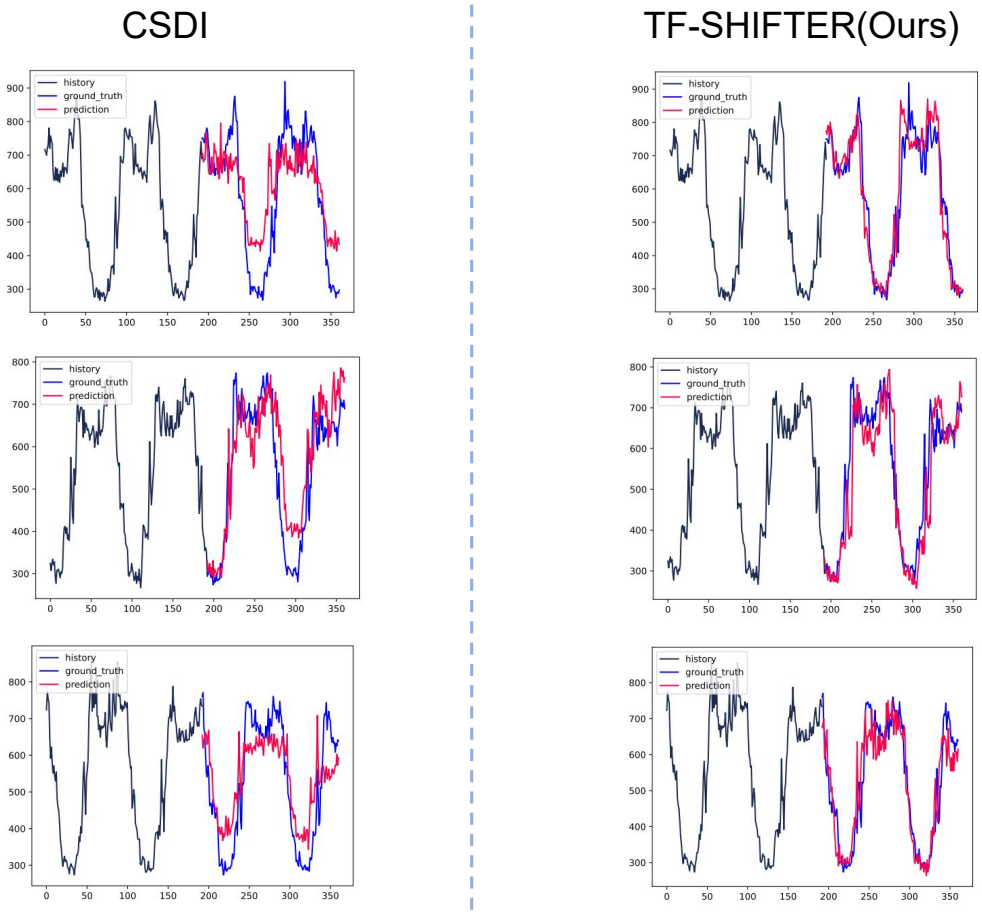

Figure 3. The comparison between TF-SHIFTER and CSDI on the electricity dataset, with our method shown on the right. In the figure, red lines represent the predicted results, blue lines represent the true values, and black lines represent the historical time series.

Noise Schedule. Our approach employs a shifting sequence k_t to determine the noise schedule in the diffusion process. Considering that the noise schedule may impact the performance of the diffusion model, we designed comparative experiments to select the noise schedule parameters. The subsequent exposition mainly revolves around the construction of the shifting sequence k_t. For the intermediate timesteps, i.e., $t \in [2, T-1]$, we propose a non-uniform geometric schedule for $\sqrt{k_t}$ as follows:

$$\sqrt{k_t} = \sqrt{k_1} \times b^{\beta_t}, t = 2, \ldots, T-1, \tag{15}$$

where

$$\beta_t = (\frac{t-1}{T-1})^p \times (T-1), b = exp[\frac{1}{2(T-1)} log \frac{k_T}{k_1}] \tag{16}$$

In order to determine the most suitable value for the parameter p in our model, we observed the results of the comparative experiments and ultimately chose 0.3 for application in the experiments.

Representation of Time Series. For the transformation from spectrum to LATENT, we conducted experiments using VQ-VAE, which was still a result obtained through comparison. We selected 1000 reconstructed sequences and their original sequences from the test set for analysis. We show t-SNE visualizations in Figure 4. VAE and TimeVAE(**?**) are two representative VAE baselines, respectively. In the figure, the synthetic samples generated from VQ-VAE are the best.

Table 3. Performance comparison of TF-SHIFTER on the *Exchange* under different p

Metric	MSE	MAE	QICE	CRPS
0.3	**0.015**	**0.073**	**4.332**	**0.339**
0.5	0.016	0.075	4.338	0.342
0.7	0.018	0.077	4.348	0.345
1.0	0.021	0.081	4.359	0.349
2.0	0.025	0.086	4.381	0.357

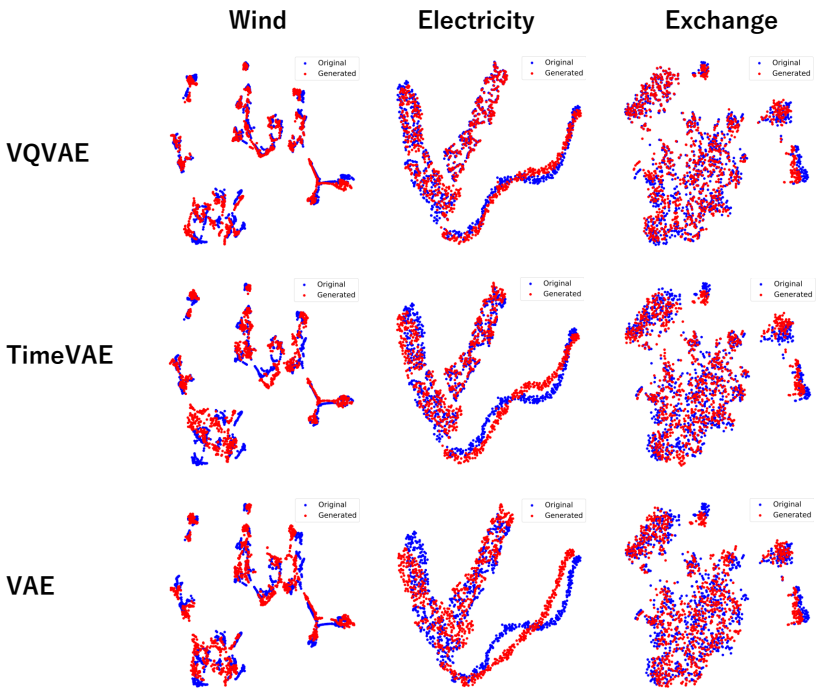

Figure 4. t-SNE plots for VQ-VAE (1st row), TimeVAE (2nd row), and VAE (3th row). Blue and red dots mean original and synthesized samples, respectively.